

Claim 1:

A graphical DSP device programming environment adapted for integration within a windows-based host application and comprising:

- (a) means for creating on a windows-based host application desktop a DSP task container panel for receiving one or more virtual panels that each represent a DSP device function and that collectively may be configured to implement a specific DSP device task;
- (b) means for creating virtual panels for placement in a said DSP task container whereby each of said virtual panels represents a selected DSP device function and has a generic property page comprising means for configuring selected parameters of said selected DSP device function;
- (c) means for creating and saving a file comprising a listing of descriptors that includes, for each virtual panel within a said DSP task container, descriptors of each DSP device function property page configuration selection made by a user in order to define a DSP device task implementation;
- (d) translation means for translating said file into an executable DSP operating system file, comprising DSP-executable modules selected from libraries of pre-compiled binary code for executing DSP functions;
- (d) means for loading said executable DSP operating system file onto a target DSP device.

Claim 2:

The DSP programming environment of claim 1 wherein said means for creating a plurality of virtual panels comprises browser means for browsing a catalog of available DSP function contained in a stored DSP operating system.

Claim 3:

The DSP programming environment of claim 2 wherein said catalog contains descriptions of available DSP functions set forth in eXtensible Markup Language (XML).

Claim 4:

The DSP programming environment of claim 3 wherein said catalog comprises, for at least one target DSP device, an XML-based description of each functional component of said at least one target DSP device.

Claim 5:

The DSP programming environment of claim 1 comprising means for creating, whenever any of said virtual panels is opened, a generic property page comprising graphical means including editor tools for enabling a user to set up selected properties of the DSP function represented by the virtual panel thus opened.

Claim 6:

The DSP programming environment of claim 5 further comprising means for creating, and associating with each of said virtual panels, a file adapted to retrievably store in memory on the host application the data representing the property page values and parameters selected by a user in order to configure the DSP function associated with a said virtual panel.

Claim 7:

The DSP programming environment of claim 1 further comprising a plurality of DSP task container panels each adapted to receive at least one DSP function panel and

to implement a specific DSP task.

Claim 8:

The DSP programming environment of claim 1, wherein said file comprising a listing of descriptors of virtual panel property page configuration selections is set forth in aspect interaction language and comprises aspect interaction language descriptors of all configured properties of all DSP function panels within a said container panel.

Claim 9:

The DSP programming environment of claim 8, wherein said aspect interaction language file of descriptors comprises sufficient DSP function property configuration descriptors to define a programmed DSP task.

Claim 10:

The DSP programming environment of Claim 9, wherein said descriptors include data representing DSP task process identification, used pipes and their variables, needed resources, used function objects and properties of said objects, and interconnections between function objects.

Claim 11:

The DSP programming environment of dependent claim 9, further comprising utility means for parsing said aspect interaction file and translating its contents into DSP-readable form.

Claim 12:

The DSP programming environment of preceding claim 11 further comprising libraries of DSP-executable binaries stored as function panels, and utility means for employing said libraries to configure said DSP-readable form of said file into parameters of binary DSP modules.

Claim 13:

The DSP programming environment of claim 12 wherein said libraries of DSP-executable binaries comprises pre-compiled binary modules sufficient to configure all user-programmable functions of a plurality of different DSP hardware devices.

Claim 14:

The DSP programming environment of claim 1 further comprising means for creating a plurality of task-oriented, custom DSP operating systems for a plurality of different DSP hardware devices, each of said custom DSP operating systems comprising DSP-executable binaries representing a set of DSP functions configured by a user.

Claim 15:

The DSP programming environment of claim 14 further comprising means for loading a said custom DSP operating system onto a selected one of a plurality of target DSP devices.

Claim 16:

The DSP programming environment of claim 1 wherein said windows-based host

application is a test and measurement application.

Claim 17:

The DSP environment of claim 7 further comprising a DSP device panel adapted to receive and to contain a plurality of said DSP task container panels, whereby said DSP device panel may be configured to implement a plurality of DSP tasks all intended for implementation on a specific DSP device.

Claim 18:

The DSP environment of claim 17 further comprising means whereby within a said DSP device panel a DSP task may be associated with at least one other DSP task.

Claim 19:

The DSP environment of claim 17 whereby said DSP device panel serves as host for device global data including global variables.

Claim 20:

A graphical development environment for programming DSP applications, said development environment comprising:

- (a) a windows-based application builder having a graphical user interface,
- (b) library files accessible by said application builder that comprise pre-compiled binary code for executing a DSP function,
- (c) means for displaying on said graphical user interface a plurality of representations of DSP functions for selection by a user,
- (d) means for enabling a user to select at least one DSP function represented

on said graphical user interface,

(e) means for enabling a user to configure at least one property of a DSP function selected by said user,

(f) means for generating a description of a user-selected and configured DSP function, and

(g) translation means for associating a said description of a user-selected and configured DSP function with at least one library file comprising pre-compiled binary code for executing said user-selected and configured DSP function on a DSP device.